

Paralelizacija optimizacije rojem čestica

Seminarski rad u okviru kursa
Naučno izračunavanje
Matematički fakultet

Milomir Radojević
milomir993@gmail.com

30. jun 2019.

Sažetak

Optimizacija rojem čestica je tehnika optimizacije inspirisana ponašanjem jata ptica u potrazi za hranom. Ova tehnika je imala dosta uspeha u rešavanju neprekidnih problema, a moguće je i prilagoditi je diskretnim problemima. Problem trgovačkog putnika je problem nalaženja Hamiltonovog ciklusa u grafu takvog da se se minimizuje suma dužina grana koje pripadaju tom ciklusu. Ovaj rad opisuje paralelni algoritam za rešavanje problema trgovačkog putnika korišćenjem tehnike optimizacije rojem čestica. Prilagođavanje ove tehnike problemu trgovačkog putnika je urađeno modifikovanjem načina predstavljanja vektora brzine i pozicije i uvođenjem novih operatora.

Sadržaj

1	Uvod	2
2	Detalji implementacije	2
3	Rezultati	4
	Literatura	4
A	Dodatak	4

1 Uvod

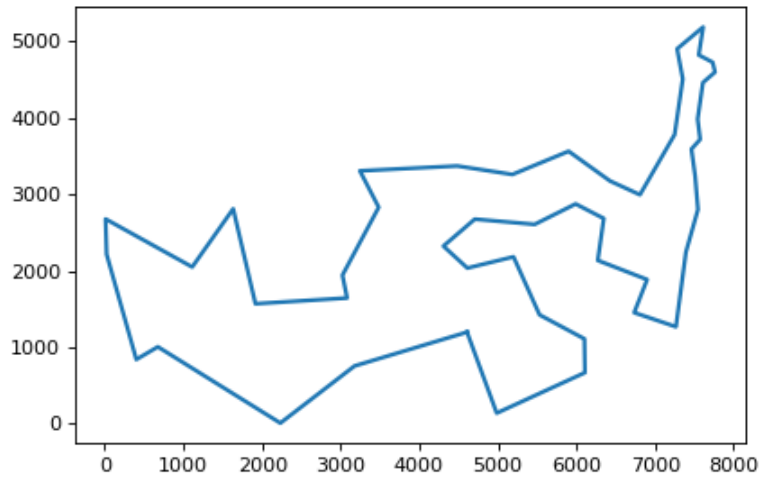
Problem trgovačkog putnika (engl. *travelling salesman problem - TSP*) je problem nalaženja najkraće putanje u grafu kojom je moguće obići sve čvorove krenuvši iz proizvoljnog čvora i vratiti se nazad u početni čvor. Ako se posmatra graf $G(N,E)$ gde su $N = \{1, \dots, n\}$ čvorovi a $E = \{1, \dots, m\}$ grane, a c_{ij} je dužina grane koja spaja čvorove i i j , problem TSP se može formulisati kao problem nalaženja Hamiltonovog ciklusa sa najmanjom ukupnom dužinom u grafu G - gde se dužina ciklusa definiše kao suma dužina svih grana. TSP ima dosta primena od kojih su neke dizajniranje mikročipova, raspoređivanje poslova, primene u bioinformatici itd.

Optimizacija rojem čestica (engl. *particle swarm optimization - PSO*) je algoritam zasnovan na inteligenciji roja. Skup mogućih rešenja problema predstavlja prostor pretrage. Individualna rešenja su predstavljena česticama. Skup čestica čini populaciju (roj). Čestice imitiraju ponašanje živih bića (npr. jato ptica) kretanjem kroz prostor pretrage. Kretanje čestica se usmerava imajući u vidu njihovu trenutnu poziciju, njihovu do sada najbolju poziciju i do sada najbolju poziciju čitavog jata. U svakoj iteraciji, preračunava se vektor brzine čestice i koristi za ažuriranje pozicije čestice. Proces se ponavlja dok se ne ispune kriterijumi za zaustavljanje. U svakoj iteraciji se ažuriraju najbolja vrednost svake pojedinačne čestice kao i celog roja. PSO se može prilagoditi diskretnim problemima na više načina od kojih je jedan modifikovanje operatora brzine i načina predstavljanja pozicije.

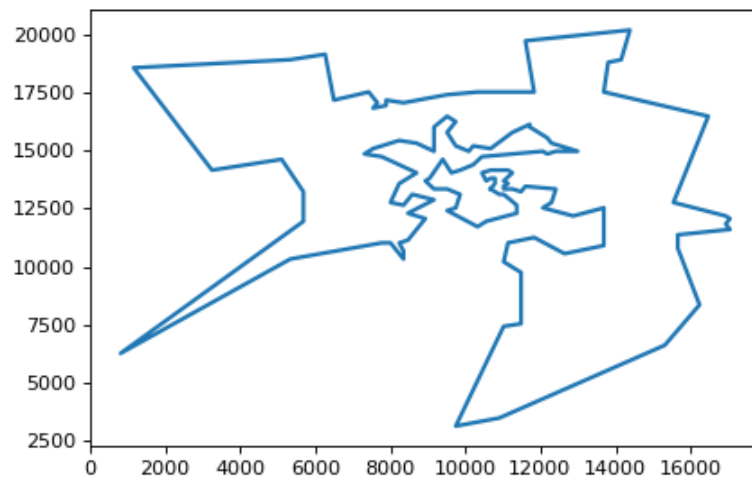
2 Detalji implementacije

```
Algoritam PSO_TSP:
  initialize()
  do:
    for each particle p:
      value(p) = evaluate(p)
      if value(p) < value(best(p)):
        best(p) = p
      if value(p) < value(global_best):
        global_best = p
    for each particle p:
      define_velocity(p)
      update_position(p)
      update_probabilities()
  while stop criterion is not satisfied
```

Pseudokod prikazuje algoritam PSO primenjen na problem TSP. U koraku inicijalizacije svakoj čestici dodeli se slučajna permutacija liste gradova. Onda se u petlji koja se izvršava dok se ne ispune kriterijumi za zaustavljanje prvo za svaku česticu pojedinačno a onda i za ceo roj ažurira najbolja vrednost, pa se za svaku česticu izračuna vektor brzine i primeni na trenutnu poziciju i na kraju ažuriraju verovatnoće koje se koriste pri računanju vektora brzine. Za vektor brzine se bira jedna od četiri operacije koje mogu da se izvrše nad jednom česticom. To su inercija, kognitivni, socijalni i kaos operatori [1]. Pod inercijom se podrazumeva lokalna pretraga korišćenjem strategije „inverznog susedstva“ (engl. *inversion neighborhood*) koja se sastoji iz invertovanja svih podnizova trenutne čestice (tj. njenog trenutnog rešenja, odnosno permutacije) i uzimanja najboljeg



Slika 1: Problem att48



Slika 2: Problem bier127

od dobijenih rezultata. Kognitivni i socijalni operatori se zasnovaju na strategiji „prevezivanja putanja“ (engl. *path relinking*) koja se sastoji iz niza zamena mesta elementima unutar jednog rešenja tako da se u od tog rešenja dođe do drugog rešenja i uzimanja najboljeg od međukoraka. U slučaju kognitivnog operatora, od trenutne pozicije čestice se kreće do najbolje prethodne pozicije te čestice, dok se u slučaju socijalnog operatora, kreće ka globalnoj najboljoj poziciji - najboljoj poziciji celog roja.

Operator haosa bira slučajnu permutaciju i njom ažurira trenutnu poziciju čestice, značajan je da bi sprečio mogućnost da sve čestice odu na istu poziciju i prestanu sa daljom pretragom. Verovatnoće za odabir svakog od različitih tipova vektora brzine se dinamički menjaju nakon svake iteracije, pri čemu je na početku najveća verovatnoća da se izvrši operator inercije, dok se ta verovatnoća smanjuje u kasnijim iteracijama. Kriterijumi za zaustavljanje su kompletiranje maksimalnog broja iteracija i maksimalnog broja iteracija bez poboljšanja globalno najboljeg rešenja.

3 Rezultati

Algoritam je primenjen na nekoliko poznatih instanci TSP-a, među kojima su att48 (slika 1), danzig42, bier127 (slika 2). Broj čestica je postavljen na 1000. Kriterijumi za zaustavljanje su 100 iteracija i 10 iteracija bez poboljšanja najbolje vrednosti. Inicijalno je najveća verovatnoća data operatoru inercije, tako da se u početnim iteracijama često vrši lokalna pretraga dok se kasnije češće vrše premošćavanja ka najboljoj (lokalnoj ili globalnoj) vrednost do tog trenutka. Za probleme att48 i danzig42 algoritam nalazi optimalno rešenje za nekoliko sekundi u velikoj većini slučajeva, dok za problem bier127, pri izvršavanju od par minuta nalazi rešenje koje se za najviše 1% razlikuje od optimalnog.

Literatura

- [1] Particle Swarm Optimization Algorithm for the Traveling Salesman Problem.

A Dodatak

Link na Github: <https://github.com/MilomirRadojevic/PSO-TSP>